Tarjeta de Control EPI
Manual de usuario

1. Introducción

Este manual le va a enseñar cómo es la Tarjeta de Control EPI, y como manejarla. Aquí no se intenta enseñar como se maneja un microcontrolador, ni tan siquiera como es el PIC16f877, básicamente se dan unos pequeños apuntes para conocer donde está conectado cada pin del microcontrolador y como conectar periféricos (sensores y actuadores).

La tarjeta de control EPI, está diseñada para ofrecer todos los recursos disponibles del microcontrolador PIC16F877, y a la vez que se pueda construir un microbot con ella, sin necesidad de hardware adicional, por lo cual tiene incorporada elementos para controlar los periféricos básicos que se utilizan en un microbot (motores y sensores).

2. Características.

A primera vista vemos que tiene un LCD (Pantalla donde se escriben cosas). Algo importante a la hora de depurar y testear, pero que no es imprescindible. También se puede observar tres pulsadores, que son: uno de reset, y los otros 2 para que los utilice el usuario a modo de comunicación burda con el cerebro del futuro robot. Además se puede apreciar un LED que se utilizará para realizar pruebas de que nuestra tarjeta funciona correctamente. Aparte de todo esto hay conectores tipo como: clemas de PCB que se utilizan para la alimentación y los motores, conectores tipo telefónico utilizados para programación y comunicaciones con el PC, y muchos conectores de 4 pines (Molex) que se utilizaran para conectar los sensores como norma general.

- **Microcontrolador PIC16F877-20/**L (En las gamas altas, hay otros que son totalmente compatibles en cuanto a patillaje se refiere, por lo tanto se podría montar directamente un micro más potente)
  - CPU RISC de arquitectura paralela, o lo que es lo mismo, que tiene diferente memoria para los datos y el programa.
  - Solo 35 instrucciones en ensamblador, que se ejecutan en un ciclo máquina, excepto las de salto que lo hacen en 2 ciclos máquina (vel_reloj / 4).
  - Velocidad máxima de operación 20 MHz ➔ Ciclo máquina 200ns
  - 8K palabras(14 bits) de memoria Flash de programa
  - 368 bytes de memoria de datos (RAM)
  - 256 bytes de memoria de datos (EEPROM)
  - 1 Watchdog
  - 2 Timer/contador de 8 bits
  - 1 Timer/contador de 16 bits
  - 2 canales de PWM para controlar motores
  - 8 canales de conversión analógico-digital de 10bits
  - Puerto serie síncrono con SPI
  - I2C
  - USART
  - Puerto Paralelo esclavo (PSP) de 8 bits

- **Alimentación PCB** por medio de reguladores lineales de tensión, configurable para que se caliente menos, según el voltaje de entrada. Seleccionable por medio de Jumper (JP1). Si está en la posición más cercana al borde del PCB, se utiliza para baterías de 6V a 10V. Si la posición es la más alejada del borde del PCB, se usa para baterías de 10V a 24V.

- **Alimentación de motores** independiente, con su controlador de motores incluido tipo puente en H con el circuito integrado L293. Además, por las características del micro, se controla por medio de PWM hardware.

- **Conexión directa de sensores a la placa:** Como ya se verá más adelante, gracias a unas resistencias integradas en la TC_EPI que se pueden cambiar mediante método de soldador y estaño, la mayoría de sensores no necesitará más que los cables y conectores para ser conectada a la placa.

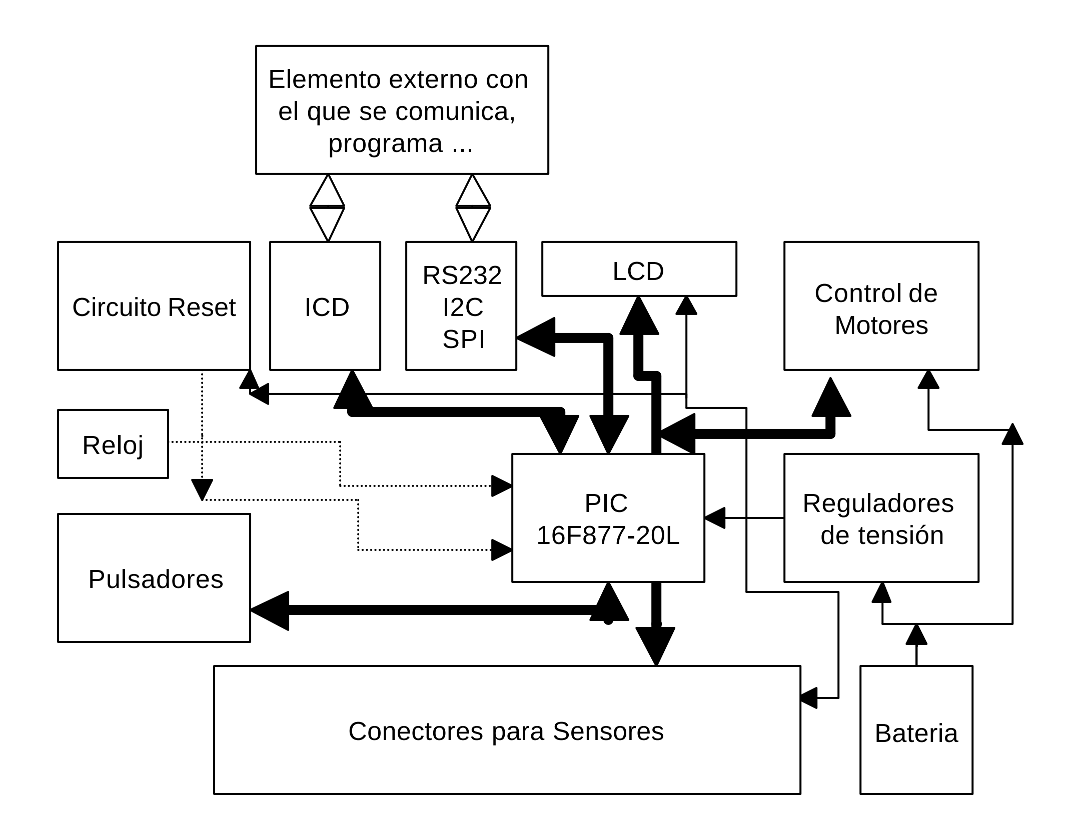
- **LCD:** Se trata de una herramienta muy vistosa (huy, me ha dicho "Hola mundo") y a la vez muy útil para testeo en el momento y herramienta de depuración de programas.

- **LED de pruebas:** Conectable y desconectable por medio del Jumper **JP2** para testeo de funcionamiento de TC_EPI y pruebas. Conectado a pin **RD1**.

- **Pulsadores integrados:** 2 pulsadores conectados a 2 líneas del micro, que se usan mucho para seleccionar un modo de programa, velocidad...

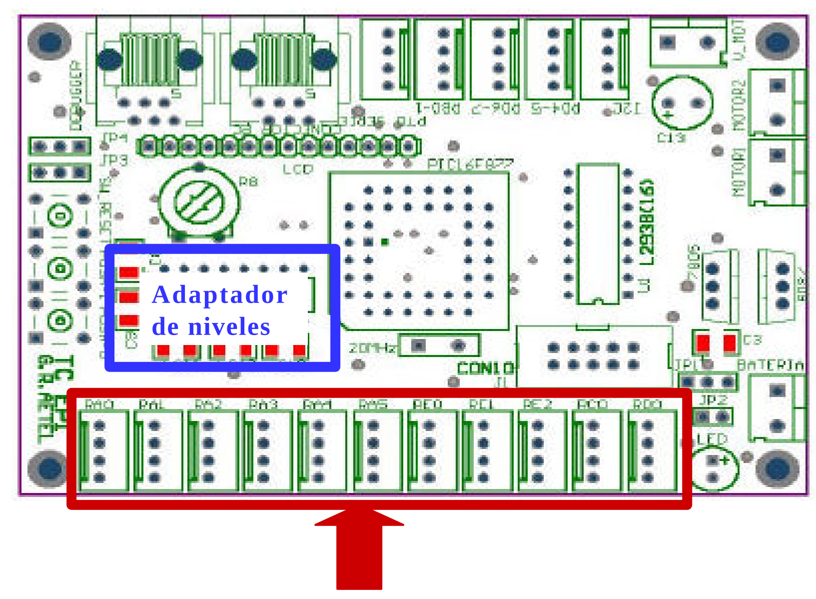
3. Aspecto Real, y localización de cada cosa

Conexiones con respecto al microcontrolador (diagrama de bloques):



3.1 Conectores Molex de 4 pines con una señal:

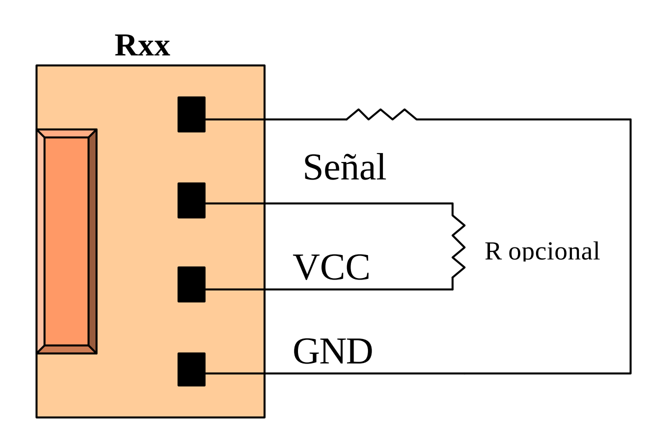
Con los nombres: RA0, RA1, RA2, RA3, RA4, RA5, RE0, RE1, RE2, RC0 y RD0, cuya señal corresponde al pin con el mismo nombre en el PIC, y por lo tanto tendrá sus funcionalidades.



Molex de 4 pines con 1 señal

¿Por qué para sacar una señal al exterior se utiliza un conector de 4 pines?

Muy sencillo. A la hora de diseñar, se ha tenido muy en cuenta conseguir conectar directamente la mayoría de sensores a la placa sin necesidad de circuitería adicional. Una vez estudiado el mercado de sensores, se ha llegado a tres casos que vendrían como anillo al dedo al diseño:



a) El sensor necesita dos líneas: Alimentación o masa, y señal. Con resistencia de pull-up en señal. Por ejemplo: Bumper de Contacto (Masa y resistencia de pull-up en señal).
b) El sensor necesita tan solo tres líneas, y son: Alimentación, señal y masa. Con resistencia de pull-up en señal. Por ejemplo: GP2D12 (Vcc, GND y señal).
c) El sensor necesita cuatro líneas, y son: Alimentación, señal, masa y una resistencia a masa. Con resistencia de pull-up en señal. Por ejemplo CNY70 (Vcc al ánodo del diodo, GND al emisor del fototransistor, resistencia a GND al katodo del diodo, y señal con resistencia a Vcc al colector del fototransistor).

En caso de que nuestro sensor necesite otro tipo de montaje (normalmente admiten varios), no queda más opción que hacer una chapucilla o montarle electrónica fuera. Si

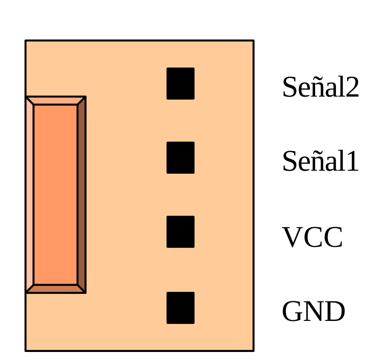
nuestro sensor necesitase 2 líneas de control, tendríamos que utilizar los conectores Molex de 4 pines con 2 señales, situados en el otro extremo de la TC_EPI.
Las resistencias se colocaran o no según la necesidad del sensor, así como sus valores.

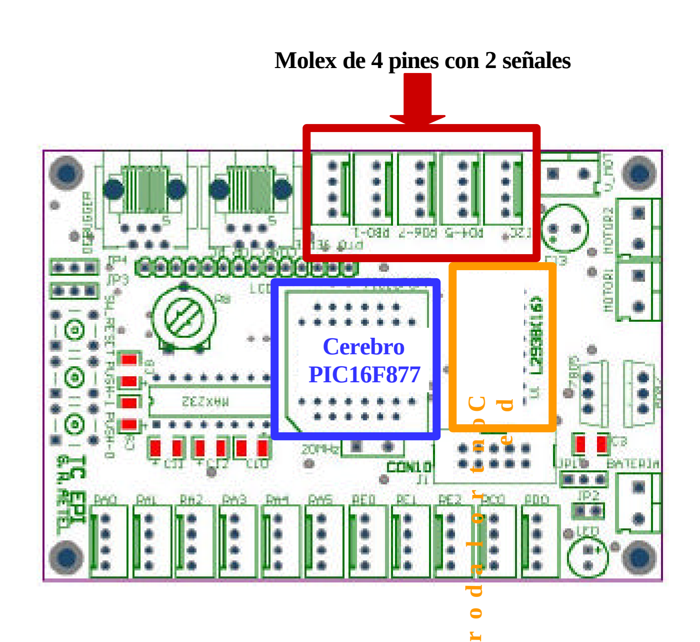
Funcionalidades de cada uno de los pines:
Si funcionan como Rxx, son simples entradas/salidas digitales.

- RA0 ➔ RA0 y AN0 (entrada analógica 0 del conversor de 10 bits)
- RA1 ➔ RA1 y AN1
- RA2 ➔ RA2, AN2 y Vref- (Como votaje de referencia negativa del ADC)
- RA3 ➔ RA3, AN3 y Vref+ (Como votaje de referencia positivo del ADC)
- RA4 ➔ RA4 (Salida en colector abierto) y T0CKI(Entrada de reloj del modulo Timer0)
- RA5 ➔ RA5, AN4 y SS (Selección esclavo para el puerto serie síncrono)
- RE0 ➔ RE0 y AN5
- RE1 ➔ RE1 y AN6
- RE2 ➔ RE2 y AN7
- RC0 ➔ RC0, T1OSO (Timer1 salida oscilador) y T1CKI (Entrada de reloj del modulo Timer1).
- RD0 ➔ RD0

3.2 Conectores Molex de 4 pines con 2 señales:

- I2C ➔ Rc3 (señal 1) y RC4 (señal 2). También se puede utilizar para comunicaciones modo bus IIC.
- RD4-5 ➔ Rd4 (señal 1) y Rd5 (señal 2)
- RD6-7 ➔ Rd6 (señal 1) y Rd7 (señal 2)
- RB0-1 ➔ Rb0 (señal 1) y Rb1 (señal 2) en paralelo con los pulsadores.
  Rb0 = Pulsador 0
  Rb1 = Pulsador 1
- PTO_SERIE ➔ Rc6 (señal 1) y Rc7 (señal 2) corresponde al puerto serie, pero sin pasar por el MAX232 (adaptador de niveles).





3.3 Motores (Compartido con J1) Y Led por medio de un Jumper

Controlados por medio de puente en H cada uno, integrado en el circuito L293.
La alimentación se realiza por medio del conector tipo clema denominado V_MOT, cuya GND está en la entrada más cercana del tornillo, y Vcc en la otra posición.

- RD1, RC5 (Dirección Motor 1; Adelante, atrás, parada rápida)
  RD1 compartido con Jumper JP2 que habilita el LED de pruebas.
- RD3, RD2 (Dirección Motor 2: Adelante, atrás, parada rápida)
- RC1 Enable Motor 1(PWM).
- RC2 Enable Motor 2(PWM).

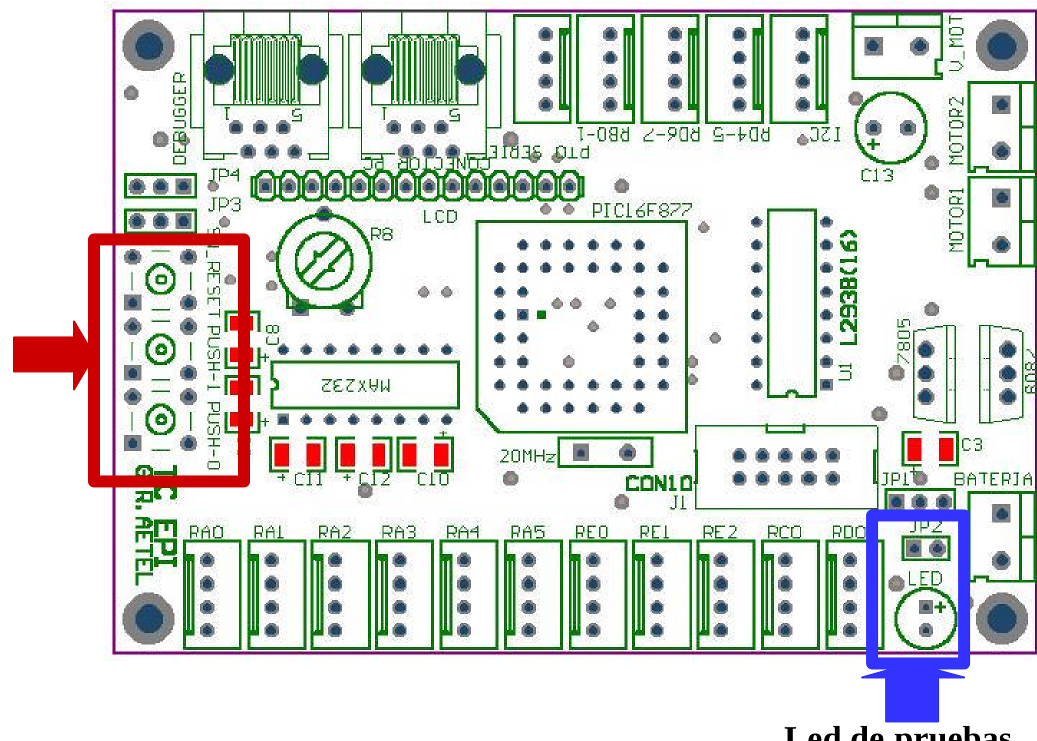
Estas líneas se ofrecen al exterior por medio del conector de 5x2: **J1**. Esto sirve para que si no se va a utilizar la parte de la placa de controlar motores, podemos usar esas líneas para conectar otra placa.

3.4 Pulsadores

- RESET ➔ Conectado con el reset del PIC
- PUSH-1 ➔ Conectado con señal RB1. Compartido con conector RB0–RB1
- PUSH0 ➔ Conectado con señal RB0. Compartido con conector RB0–RB1

3.5 Led de pruebas

Se utiliza una señal de los motores para conectar el LED de la placa. Este sirve pata comprobar que funciona bien. Se conecta y desconecta por medio del Jumper JP2, y va a la línea RD1.



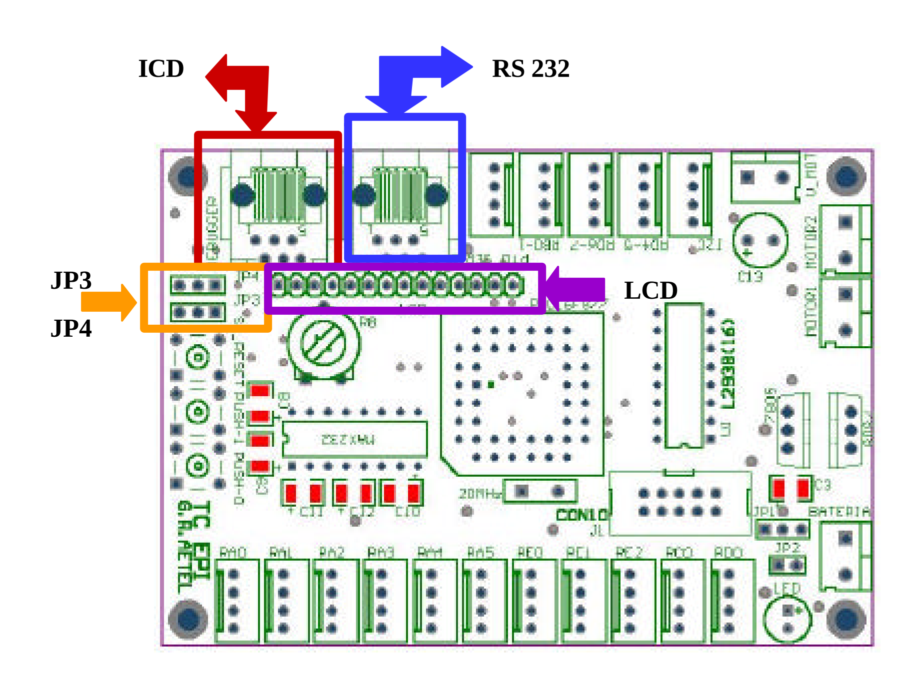
Led de pruebas

3.6 LCD

El LCD está integrado en la placa, de forma que es como un elemento más.
El modo de utilización es a bus multiplexado de 4 líneas, y se controla con un total de 6.

Pin del LCD → Conexión en TC_EPI

- Pin1 (Vss) → GND
- Pin2 (Vcc) → VCC
- Pin3(Vc) → VCC
- Pin4(RS) → RB3 compartida con conector RJ11: DEBUGGER
- Pin5(R/W) → GND
- Pin6(E) → RB2
- Pin7(D0) → Nada (al aire)
- Pin8(D1) → Nada (al aire)
- Pin9(D2) → Nada (al aire)
- Pin10(D3) → Nada (al aire)
- Pin11(D4) → RB4
- Pin12(D5) → RB5
- Pin13(D6) → RB6_A. RB6 si Jumper JP3 está posición alejada del borde del PCB.
- Pin14(D7) → RB7_A. RB7 si Jumper JP4 está posición alejada del borde del PCB.



3.7 RS_232 (Conector PC)

Conector tipo RJ11. En TC_EPI se denomina conector PC, y es el RJ11 más cercano de los conectores tipo molex de 4 pines.
Este utiliza el adaptador de niveles MAX232, que conectado a lar líneas del PIC RC6 y RC7 es un puerto RS232 estándar, con las líneas de TX, RX y GND.
Se conectará al PC por medio de un cable tipo telefónico de 4 o 6 hilos, y un adaptador RJ11-DB9.

3.8 Debuger (ICD)

Conector tipo RJ11. Denominado DEBUGGER, y se usa para conectar un programador de PICs o el ICD. Para Poder utilizarlo, hay que conectar JP3 y JP4 en la posición cercana al borde del PCB.

3.9 Alimentación

Las baterías se conectarán en el conector con el nombre BATERIA, cuya entrada de negativa (GND) se corresponde con el pin más cercano al tornillo, y el otro con el positivo.

Para una menor disipación del regulador, se ha colocado el Jumper JP1, pudiendo alimentar el resto de la circuitería por medio de un regulador, no quedando como opción la de alimentar directamente la circuitería desde la batería. Si la batería es de entre 6 y 10 voltios, se recomienda la colocación de JP1 en la posición más cercana del borde del PCB. Si la batería fuese entre 10 y 24 voltios, se recomienda la posición más alejada, para no dañar el primer regulador de tensión.

